UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/721,695	11/24/2000	Matthew A. Feinberg	C24-002	4384

28156 7590 08/26/2004

COLEMAN SUDOL SAPONE, P.C.
714 COLORADO AVENUE
BRIDGE PORT, CT 06605-1601

EXAMINER	
TANG, KENNETH	

ART UNIT	PAPER NUMBER
2127	

DATE MAILED: 08/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/721,695	FEINBERG, MATTHEW A.	
	Examiner	Art Unit	
	Kenneth Tang	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11/24/00* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *10/7/03*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-45 are presented for examination.

Drawings

2. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the "modifying an object on said display in response to instructions specified by a respective active virtual thead in said second linked list" (independent claim 21) must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

3. Claim 44 is objected to because of the following informalities: "byte-" should be changed to "byte" (in page 57, lines 20 and page 58, line 5). Appropriate correction is required.

Claim Rejections - 35 USC § 112

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 1-45 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention:

a. The following terms lack antecedent basis:

i. "the computer" – claim 41, lines 16-17

ii. "local variables" – claim 44, lines 17-18

iii. "said virtual program threads" – claim 45, line 14

b. The following claim languages are indefinite:

iv. In claim 1, the term "respective" (lines 10 and 11) is indefinite because it is not made explicitly clear in the claim language what this is respective to. It is also not made explicitly clear in the claim language whether "a respective one"

(line 11) is the same as the "respective virtual thread" (line 7) or if a separate respective one is created.

v. Claims 21, 31, 41 and 44 are rejected for the same indefinite reasons as stated above in the rejection of claim 1.

vi. In claim 21, "an object" (line 9) is indefinite because it is not made explicitly clear in the claim language whether this refers to an item or a data software object.

vii. In claim 44, "interval" (line 9) is indefinite because it is not made explicitly clear in the claim language how this distinguishes from a "time slice" (line 4), if there is a difference between those terms.

viii. In claim 45, "objects" (line 14) is indefinite because it is not made explicitly clear in the claim language whether this refers to an item or a data software object.

ix. In claim 21, it is indefinite because there is no structural relationship established between the memory and the interpreter. Specifically, it is not made explicitly clear what the bytecode or pseudocode instructions have to do with the state and context data of the virtual threads because a structural relationship has not been established between them.

x. In claim 41, it is indefinite because there is no structural relationship established between the memory and the interpreter.

xi. In claim 44, it is indefinite because there is no structural relationship established between the memory and the interpreter. Specifically, it is not made

explicitly clear what the byte or pseudocode instructions have to do with the state and context data of a current one of said virtual threads because a structural relationship between the these feature of the interpreter and memory had not been established.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-18, 20, and 31-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bitar et al. (hereinafter Bital) (US 6,766,515 B1) in view of Diepstraten et al. (hereinafter Diepstraten) (US 6,260,150 B1).

6. As to claim 1, Bitar teaches a method for operating a computer, comprising:

storing in a computer memory a plurality of pseudocode instructions, at least some of said pseudocode instructions comprising a plurality of machine code instructions *(col. 6, lines 9-12 and 43-65, col. 5, lines 34-41)*;

for each of a plurality of tasks or jobs to be performed by the computer, automatically creating a respective virtual thread of execution context data including (a) a memory location of a next one of said pseudocode instructions (thread to run in next time slice) to be executed in carrying out the respective task or job (mapped) and (b) the values of any local variables required

for carrying out the respective task or job, a plurality of said tasks or jobs each entailing execution of a respective one of paid pseudocode instructions comprising a plurality of machine language instructions; processing tasks or jobs in a respective series of time slices or processing slots under the control of the respective virtual thread; and context switching between threads *(col. 1, lines 25-35, col. 4, lines 10-20 and 57-67, col. 5, lines 19-25, col. 6, lines 9-12, col. 13, lines 30-49)*.

7. Bitar fails to explicitly teach performing context switching only after completed execution of a currently executing one of said pseudocode instructions. However, Diepstarten teaches time slice context switching to occur at the end of every instruction *(col. 8, lines 42-49 and 63-66)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of time slice context switching to occur at the end of every instruction to the existing system of Bitar because it is preferred to switch states or instructions when the instruction is over/completed *(col. 8, lines 50-66)*.

8. As to claim 2, Bitar teaches wherein each of the virtual threads is part of a respective linked list of virtual threads, each of the virtual threads further including a pointer to a next virtual thread (thread to run in next time slice) in the respective linked list, further comprising, for every context switch between different virtual threads, consulting the pointer of a currently executing virtual thread to determine an identity of a next virtual thread to be executed *(col. 13, lines 30-49, col. 16, lines 5-7)*.

9. As to claim 3, Bitar teaches wherein said respective linked list is one of a plurality of linked lists of said virtual threads, one of said linked lists being a list of idle virtual threads, another of said linked lists being a list of active virtual threads, an additional one of said linked lists being a list of queued virtual threads, further comprising periodically moving at least one virtual thread from said list of queued virtual threads to said list of active virtual threads *(col. 11, lines 1-7, col. 12, lines 7-21)*.

10. As to claim 4, Bitar teaches wherein the moving of a virtual thread from said list of queued virtual threads to said list of active virtual threads includes: setting a mutex to lock said list of queued virtual threads; subsequently modifying pointers in (i) the moved virtual thread, (ii) at least one virtual thread originally in said list of active virtual threads, and (iii) at least one virtual thread remaining in said list of queued virtual threads; and thereafter resetting or releasing the mutex to enable access to said list of queued virtual threads *(col. 12, lines 63-67 through col. 13, lines 1-15, col. 15, lines 30-44, col. 11, lines 21-49)*.

11. As to claim 5, it is rejected for the same reasons as stated in the rejection of claim 4.

12. As to claim 6, it is rejected for the same reasons as stated in the rejection of claim 4.

13. As to claim 7, it is rejected for the same reasons as stated in the rejection of claim 2.

14. As to claim 8, Bitar teaches wherein each of said virtual threads is assigned a message queue, further comprising entering a message in a message queue of a selected one of said virtual threads during execution of a task or job pursuant to another one of said virtual threads *(col. 17, lines 22-35 and col. 18, lines 21-41)*.

15. As to claim 9, Bitar teaches wherein said selected one of said virtual threads and said another one of said virtual threads correspond to respective tasks or jobs derived from different applications programs, whereby the entering of said message in the message queue of said selected one of said virtual threads implements data transfer between said different applications programs *(col. 17, lines 22-35 and col. 18, lines 21-41)*.

16. As to claim 10, Bitar teaches wherein said selected one of said virtual threads and said another one of said virtual threads are proxy or interface threads on different computers, the entering of said message in said message queue including transmitting said message over a communications link between said computers *(col. 8, lines 28-31)*.

17. As to claim 11, it is rejected for the same reasons as stated in the rejection of claim 1. In addition, Bitar teaches using an interpreter program *(col. 1, lines 26-35)*.

18. As to claim 12, it is rejected for the same reasons as stated in the rejection of claim 1. In addition, Bitar teaches a native thread that a virtual thread is mapped from *(col. 13, lines 30-49)*.

19. As to claims 13 and 14, Bitar and Diepstraten fails to explicitly teach determining an average load of all the native threads and shifting a virtual thread from a first native thread having a heavier-than-average load to a second native thread having a lighter-than-average load. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of calculating the average load and shifting the larger avereage load to a lighter load because it is well known that this balance loading will increase the speed and efficiency.

20. As to claim 15, it is rejected for the same reasons as stated in the rejection of claim 10.

21. As to claim 16, it is rejected for the same reasons as stated in the rejection of claim 10.

22. As to claim 17, it is rejected for the same reasons as stated in the rejection of claims 3 and 8.

23. As to claim 18, Bitar teaches wherein each of said virtual threads additionally includes a thread priority, further comprising automatically consulting the thread priorities in a plurality of said virtual threads to determine relative priorities and varying a sequence of threads in accordance with the determined relative priorities *(col. 10, lines 23-29)*.

24. As to claim 20, Bitar teaches wherein said time slots or processing slots are measured by counting consecutively executed pseudocode instructions, further comprising, for each of a

plurality of said time slices or processing slots, terminating the respective time slot or processing slot upon counting a predetermined number of consecutively executed pseudocode instructions (time quantum expires) *(col. 2, lines 22-34)*.

25. As to claim 31, Bitar teaches a computer having an interpreter for executing a series of bytecode instructions each consisting of a multiplicity of machine code steps, a multitasking method comprising:

for each task of a plurality of tasks to be performed by the computer, using the interpreter to define a respective (mapped) virtual thread *(col. 1, lines 26-35)*;

during each time slice of a series of consecutive time slices, executing bytecode instructions of a respective current thread (choose to schedule an alternate thread) selected from among the virtual threads *(col. 11, lines 1-19, col. 13, lines 22-61)*; and

executing a context switch from one of said virtual threads to another of said virtual threads only after execution of one of said bytecode instructions *(col. 2, lines 22-40)*.

26. Bitar teaches virtual threading and context switching *(col. 1, lines 26-40, col. 13, line 61)* but fails to explicitly teach performing context switching only after completed execution of a currently executing one of said pseudocode instructions. However, Diepstarten teaches time slice context switching to occur at the end of every instruction *(col. 8, lines 42-49 and 63-66)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of time slice context switching to occur at the end of every instruction to the existing system of Bitar because it is preferred to switch states or instructions when the instruction is over/completed *(col. 8, lines 50-66)*.

27. As to claim 32, it is rejected for the same reasons as stated in the rejection of claim 2.

28. As to claim 33, it is rejected for the same reasons as stated in the rejection of claim 3.

29. As to claim 34, it is rejected for the same reasons as stated in the rejection of claim 4.

30. As to claim 35, it is rejected for the same reasons as stated in the rejection of claim 5.

31. As to claim 36, it is rejected for the same reasons as stated in the rejection of claim 6.

32. As to claim 37, it is rejected for the same reasons as stated in the rejection of claim 8.

33. As to claim 38, it is rejected for the same reasons as stated in the rejection of claim 10.

34. As to claim 39, it is rejected for the same reasons as stated in the rejection of claim 18.

35. As to claim 40, it is rejected for the same reasons as stated in the rejection of claim 11.

36. As to claim 41, it is rejected for the same reasons as stated in the rejection of claim 31. In addition, Bitar teaches a memory storing state and context data of multiple threads or tasks *(col. 6, lines 9-12 and 43-65, col. 5, lines 34—41, and col. 11, lines 1-20)*.

37. As to claim 42, it is rejected for the same reasons as stated in the rejection of claim 2.

38. As to claim 43, it is rejected for the same reasons as stated in the rejection of claim 3.

39. **Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bitar et al. (hereinafter Bital) (US 6,766,515 B1) in view of Diepstraten et al. (hereinafter Diepstraten) (US 6,260,150 B1), and further in view of Blanset et al. (hereinafter Blanset) (US 4,744,048).**

40. As to claim 19, Bitar in view of Diepstraten fails to explicitly teach controlling objects imaged on a computer display and monitoring actuation of keys on a computer keyboard. However, Blanset teaches controlling objects imaged on a computer display and monitoring the input of keys on a keyboard *(col. 13, lines 45-60 and see Abstract)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of controlling objects imaged on a computer display and monitoring the input of keys on a keyboard to the existing system of Bitar and Diepstraten because it would allow for context switching by the user *(col. 1, lines 6-26)*.

41. **Claims 44 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bitar et al. (hereinafter Bital) (US 6,766,515 B1) in view of Blanset et al. (hereinafter Blanset) (US 4,744,048).**

42. As to claim 44, Bitar teaches a computer method comprising:

compiling input user source code into byte- or pseudocode instructions each corresponding to a multiplicity of machine code instructions *(col. 5, lines 32-57)*;

operating an interpreter of said computer to assign computing tasks to respective (mapped) virtual threads, the assigning of said computing tasks to said virtual threads including identifying and storing state and context data for each of said computing tasks *(col. 1, lines 25-35, col. 4, lines 57-67, col. 5, lines 19-25, and col. 6, lines 9-12)*;

additionally operating said interpreter to execute selected ones of said byte- or pseudocode instructions pursuant to the state and context data of a current one of said virtual threads *(col. 11, lines 1-19)*;

after the execution of each successive one of the selected byte- or pseudocode instructions and only after such execution, further operating said interpreter to check whether a predetermined interval has elapsed since a commencement of execution of instructions pursuant to said current one of said virtual threads *(col. 2, lines 22-34)*; and

upon a determination of elapsing of said predetermined interval, operating said interpreter to perform a context switch (choose to schedule an alternate thread for some period of time) *(col. 2, lines 35-40)*.

43. As stated above, Bitar teaches operating said interpreter to execute selected ones of said byte- or pseudocode instructions pursuant to the state and context data of a current one of said virtual threads *(col. 11, lines 1-19)* but Bitar fails to explicitly teach that there is a timer to generate a series of time slices to perform this. However, Blanset teaches that timers that keep track of the timing of the time slices *(col. 13, lines 48-52)*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to include the feature of timers because it increases the control of the system *(col. 13, lines 42-59)*.

44. As to claim 45, Bitar teaches the tasks assigned to respective ones of said virtual program threads, a network, and calculating local variables *(col. 1, lines 14-35, col. 3, line 50, col. 8, line 30)*. Blanset teaches (a) controlling objects appearing in an image on a display screen, (b) monitoring operator input, (c) executing routines of applications programs, (d) running computer maintenance routines *(col. 13, lines 45-60)*.

Allowable Subject Matter

45. Claims 21-30 would be allowable if rewritten or amended to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth Tang whose telephone number is (571) 272-3772. The examiner can normally be reached on 8:30AM - 6:00PM, Every other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kt
8/16/04

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100